Software Development Principles

Lecture 1
Lists – 2 Dimensional

Lecturer:

Karen Nolan

karen.nolan@it-tallaght.ie

Lists – 2 Dimensional

- We have so far seen 1D Lists

```
myList = [1, 19, 27, 8, 5, 9]

for item in myList:
    print(item)
```

- A single list is created using [element0, element1, element2]
- A 2 dimensional list in Python is simply: ***Lists inside a list***

Lists – 1D - Revisited

- We have so far seen 1D Lists

```
myList = [2, 19, 27, 8, 5, 9]

Where:
      myList[0] = 2
      myList[1] = 19
      myList[2] = 27
      myList[3] = 8
      myList[4] = 5
      myList[5] = 9
```

Lists – 1D - Revisited

- We have so far seen 1D Lists

```
myList = [2, “karen”, 4.7]

Where:
      myList[0] = 2
      myList[1] = “karen”
      myList[2] = 4.7
```

- Remember List types can change within a list.

Lists – 2D Introduction

- A 2D list is: *Lists inside a List*

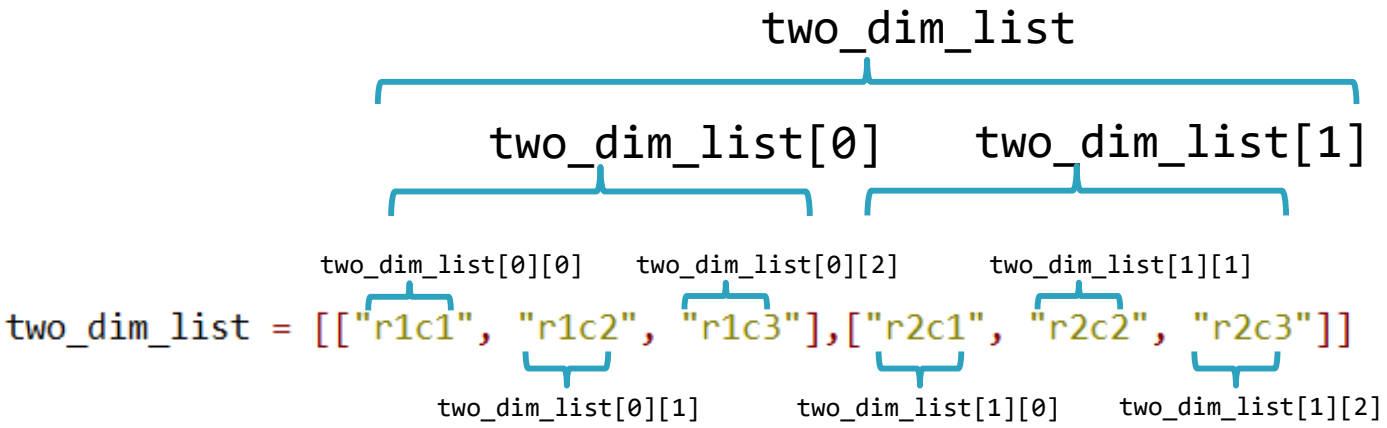
```
myList = [[23, 3, 1, 4] , [5, 22, 6, 3]]

Where:
      myList[0] = [23, 3, 1, 4]
      myList[1] = [5, 22, 6, 3]

      myList[0][0] = 23
      myList[0][3] = 4
      myList[1][1] = 22
      myList[1][2] = 23
```

Lists – 2D Introduction

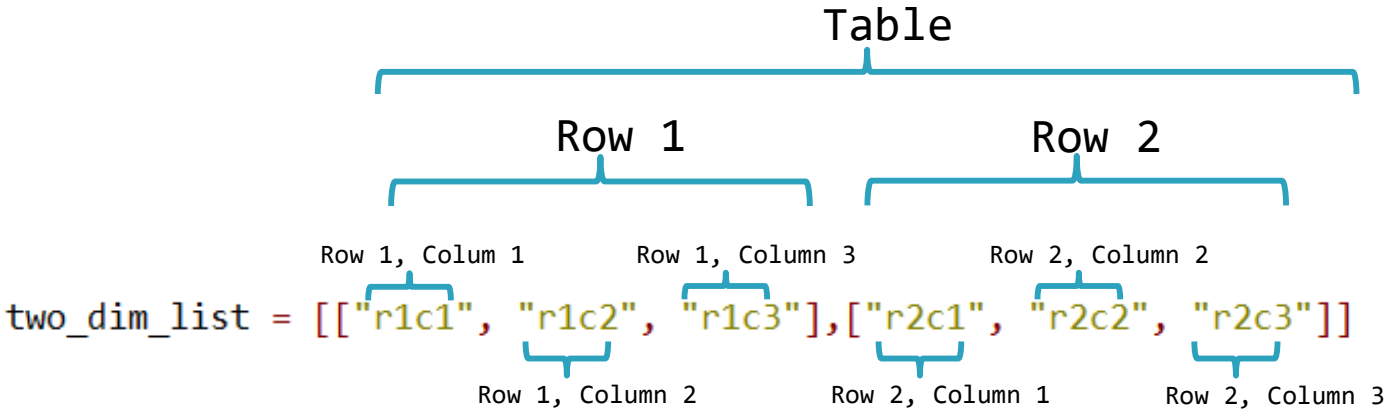
- A 2D list is: *Lists inside a List*



	0	1	2
0	r1c1	r1c2	r1c3
1	r2c1	r2c2	r2c3

Lists – 2D Introduction

- A 2D list is: *Lists inside a List*



	0	1	2
0	r1c1	r1c2	r1c3
1	r2c1	r2c2	r2c3

Lists – 2D Introduction

- A 2D list is: *Lists inside a List*

```
my2D_List = [["Honda", "Civic", "99-D-1234"],
             ["Nissan", "Pulsar", "01-KE-33456"]]

print(my2D_List)
print(my2D_List[0])
print(my2D_List[1][1])
```

- What will the output be?

Lists – 2D Introduction

- A 2D list is: *Lists inside a List*

```
my2D_List = [["Honda", "Civic", "99-D-1234"],
["Nissan", "Pulsar", "01-KE-33456"]]

1) print(my2D_List)
2) print(my2D_List[0])
3) print(my2D_List[1][1])
```

1) `[['Honda', 'Civic', '99-D-1234'], ['Nissan', 'Pulsar', '01-KE-33456']]`
2) `['Honda', 'Civic', '99-D-1234']`
3) `Pulsar`

Class Example 1: My Cars

```
my2D_List = [["Honda", "Civic", "99-D-1234"], ["Nissan",
"Pulsar", "01-KE-33456"]]
```

- Given the above list, print it like a table:

```
Make        Model       Year
-------------------------------
Honda       Civic       99-D-1234
Nissan      Pulsar      01-KE-33456
```

Class Example 1: My Cars : Solution

```
my2D_List = [["Honda", "Civic", "99-D-1234"], ["Nissan",
"Pulsar", "01-KE-33456"]]

print("{0:<10}".format("Make") + "{0:<10}".format("Model") +
"{0:<10}".format("Year"))
print("-------------------------------")
for row in my2D_List:
    for col in row:
        print("{0:<10}".format(col), end="")
    print()
```

Class Example 2: My Cars

```
my2D_List = [["Honda", "Civic", "99-D-1234"], ["Nissan",
"Pulsar", "01-KE-33456"]]
```

Given the above list:

- Add a car (using user input)
- Print it like a table:

```
my2D_List = [["Honda", "Civic", "99-D-1234"], ["Nissan",
"Pulsar", "01-KE-33456"]]

# Take in details
carMake = input("Please enter the make            :")
carModel = input("Please enter the model           :")
carReg = input("Please enter the registration   :")

# Build single List
singleCar = [carMake, carModel, carReg]

# add List to my2D_List
my2D_List.append(singleCar)

print("{0:<10}".format("Make") + "{0:<10}".format("Model") +
"{0:<10}".format("Year"))
print("---------------------------------")
for row in my2D_List:
    for col in row:
        print("{0:<10}".format(col), end="")
    print()
```

Class Example 3: My Cars

```
my2D_List = [["Honda", "Civic", "99-D-1234"], ["Nissan",
"Pulsar", "01-KE-33456"]]
```

Given the above list:

- Add 5 cars (using user input)
- Print it like a table:

Class Example 3: My Cars : Solution

```
my2D_List = [["Honda", "Civic", "99-D-1234"], ["Nissan",
"Pulsar", "01-KE-33456"]]

for i in range(5):
    carMake = input("Please enter the make           :")
    carModel = input("Please enter the model          :")
    carReg = input("Please enter the registration   :")

    singleCar = [carMake, carModel, carReg]

    my2D_List.append(singleCar)


print("{0:<10}".format("Make") + "{0:<10}".format("Model") +
"{0:<10}".format("Year"))
print("-------------------------------")
for row in my2D_List:
    for col in row:
        print("{0:<10}".format(col), end="")
    print()
```

Class Example 4: Football

- Create a program that askes the user to enter 4 football teams
- Each football team contains the following information:
  - **Name**
  - **Wins**
  - **Draws**
  - **Loses**
- Print the league table, including the teams points:
  - Win = 3 points
  - Draw = 1 points
  - Lose = 0 points

```
Team      Wins      Draws     Loses     Points
-----------------------------------------------
Man U     3         2         1         11
Liverpool 2         2         2         8
Spurs     1         2         3         5
Leeds U   0         3         3         3
```

Class Example 4: Football : Solution

```
leagueTable = []

for counter in range(4):
    name = input("Please enter team name:")
    wins = int(input("please enter the number of wins:"))
    draws = int(input("please enter the number of draws:"))
    loses = int(input("please enter the number of loses:"))
    team = [name, wins, draws, loses]
    leagueTable.append(team)

print("{0:<10}".format("Team") + "{0:<10}".format("Wins") +
"{0:<10}".format("Draws")
          + "{0:<10}".format("Loses") + "{0:<10}".format("Points"))
print("-----------------------------------------")
for team in leagueTable:
    for information in team:
        print("{0:<10}".format(information), end="")

    points = (3 * team[1]) + (1 * team[2])
    print("{0:<10}".format(points))
```

Max and Min

```
leagueTable = [["Man U", 3, 2, 1], ["Liverpool", 2, 2, 2],
["Spurs", 1, 2, 3], ["Leeds", 0, 3, 3]]
```

- We can no longer just use max(leagueTable)
- What will the below code output?

```
print(max(leagueTable))
print(min(leagueTable))
```

Max and Min

```
leagueTable = [["Man U", 3, 2, 1], ["Liverpool", 2, 2, 2],
["Spurs", 1, 2, 3], ["Leeds", 0, 3, 3]]
```

- We can no longer just use max(leagueTable)
- What will the below code output?

```
print(max(leagueTable))
print(min(leagueTable))
```

```
['Spurs', 1, 2, 3]
['Leeds', 0, 3, 3]
```

- Why did it produce this output?

Max and Min

```
leagueTable = [["Man U", 3, 2, 1], ["Liverpool", 2, 2, 2],
["Spurs", 1, 2, 3], ["Leeds", 0, 3, 3]]
```

- We must use:

```
maxWins = max([row[1] for row in leagueTable])
```

- The code produces a single List based on a single column, in this case the second column, column 1.

```
[row[1] for row in leagueTable]
```

Max and Min

```
leagueTable = [["Man U", 3, 2, 1], ["Liverpool", 2, 2, 2],
["Spurs", 1, 2, 3], ["Leeds", 0, 3, 3]]
```

- We must use:

```
[row[1] for row in leagueTable]
```

- This produces a single List, from column 1.

- **You could also loop through the List and find the largest Value:**

Max and Min

```
leagueTable = [["Man U", 3, 2, 1], ["Liverpool", 2, 2, 2],
["Spurs", 1, 2, 3], ["Leeds", 0, 3, 3]]
```

- We must use:

```
[row[1] for row in leagueTable]
```

- Or:

```
maxValueWins = leagueTable[0][1]

for team in leagueTable:
    if team[1] > maxValueWins:
        maxValueWins = team[1]

print(maxValueWins)
```

Max and Min

```
leagueTable = [["Man U", 3, 2, 1], ["Liverpool", 2, 2, 2],
["Spurs", 1, 2, 3], ["Leeds", 0, 3, 3]]
```

- Useful to get index of winning team:

```
maxValueWins = leagueTable[0][1]
indexMostWins = 0

for index, team in enumerate(leagueTable):
    if team[1] > maxValueWins:
        maxValueWins = team[1]
        indexMostWins = index

print(maxValueWins)
print(leagueTable[indexMostWins][0])
```

Stock With a Menu

- Write a program that has a menu:
  - Add stock
  - Stock list
- You will need to be able to go back to the menu
- Stock must contain an ID, description, sale price and Qty

Stock - Hint

```
menuOption = 0
stock = []

while menuOption != 5:
    print("\t************************")
    print("\t*       Menu           *")
    print("\t************************")
    print("\t* 1) Add Stock         *")
    print("\t* 2) Stock List        *")
    print("\t************************")
    print("\t* 3) Exit              *")
    print("\t************************")

    menuOption = int(input("\tPlease enter menu option:"))

    if menuOption == 1:
        pass
    elif menuOption == 2:
        pass
    elif menuOption == 3:
        pass
    else:
        print("Error - Please enter number between 1 and 3.")
```

Solution

```
menuOption = 0
stock = []
while menuOption != 5:
    print("\t**********************")
    print("\t*       Menu         *")
    print("\t**********************")
    print("\t* 1) Add Stock       *")
    print("\t* 2) Stock List      *")
    print("\t**********************")
    print("\t* 3) Exit            *")
    print("\t**********************")

    menuOption = int(input("\tPlease enter menu option:"))

    if menuOption == 1:
        print("\t**********************")
        print("\t*   Add Stock        *")
        print("\t**********************")

        stockID = input("\tPlease enter stock ID:")
        description = input("\tPlease enter stock description:")
        salePrice = float(input("\tPlease enter sale price:"))
        qty = int(input("\tPlease enter quantity:"))
        newStockItem = [stockID, description, salePrice, qty]
        stock.append(newStockItem)

    elif menuOption == 2:
        print("\t**********************")
        print("\t*   Stock  List      *")
        print("\t**********************")
        print("\t-----------------------------------------")
        print("\t{0:<10}".format("ID") + "{0:<10}".format("Des") + "{0:<10}".format("RRP") + "{0:<10}".format("QTY"))
        print("\t-----------------------------------------")
        for stockItem in stock:
            print("\t", end="")
            for detail in stockItem:
                print("{0:<10}".format(detail), end="")
            print()
    elif menuOption == 3:
        pass
    else:
        print("Error - Please enter number between 1 and 3.")
```